


```
RRRRRRRR      EEEEEEEEEEE NN      NN      AAAAAA      MM      MM      EEEEEEEEEEE
RRRRRRRR      EEEEEEEEEEE NN      NN      AAAAAA      MM      MM      EEEEEEEEEEE
RR      RR      EE      NN      NN      AA      AA      MMMM      MMMM      EE
RR      RR      EE      NN      NN      AA      AA      MMMM      MMMM      EE
RR      RR      EE      NNNN      NN      AA      AA      MM      MM      EE
RR      RR      EE      NNNN      NN      AA      AA      MM      MM      EE
RRRRRRRR      EEEEEEEEEEE NN      NN      AA      AA      MM      MM      EEEEEEEEE
RRRRRRRR      EEEEEEEEEEE NN      NN      AA      AA      MM      MM      EEEEEEEEE
RR      RR      EE      NN      NN      AA      AA      MM      MM      EE
RR      RR      EE      NN      NN      AA      AA      MM      MM      EE
RR      RR      EE      NNNN      NN      AAAAAAAAAA      MM      MM      EE
RR      RR      EE      NNNN      NN      AAAAAAAAAA      MM      MM      EE
RR      RR      EE      NN      NN      AA      AA      MM      MM      EE
RR      RR      EE      NN      NN      AA      AA      MM      MM      EE
RR      RR      EEEEEEEEEEE NN      NN      AA      AA      MM      MM      EEEEEEEEE
RR      RR      EEEEEEEEEEE NN      NN      AA      AA      MM      MM      EEEEEEEEE
```

....
....
....
....

```
LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS
```

```

1 0001 0 MODULE RENAME ( %TITLE 'VAX/VMS RENAME Utility'
2 0002 0 IDENT = 'V04-000',
3 0003 0 MAIN = RENAME_MAIN
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 * ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 * TRANSFERRED.
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 * CORPORATION.
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1
31 0031 1 **
32 0032 1 FACILITY: VAX/VMS RENAME Utility
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1 This utility renames files and directories.
37 0037 1
38 0038 1 ENVIRONMENT: VAX/VMS Operating System, non-privileged user mode
39 0039 1
40 0040 1 AUTHOR: Steven B. Lionel, CREATION DATE: 04-Oct-1982
41 0041 1
42 0042 1 Adapted from RENAME V03-000, written by Greg Robert
43 0043 1
44 0044 1 MODIFIED BY:
45 0045 1
46 0046 1 03-004 BLS0282 Benn Schreiber 5-MAR-1984
47 0047 1 Use new argument to lib$delete_file for related file
48 0048 1 processing.
49 0049 1
50 0050 1 3B-003 SBL3003 Steven B. Lionel 11-Feb-1983
51 0051 1 /UIC is now /BYOWNER.
52 0052 1
53 0053 1 3B-002 SBL3002 Steven B. Lionel 19-Jan-1983
54 0054 1 Remove special case RMSS_FEX code since $RENAME now
55 0055 1 correctly handles this.
56 0056 1
57 0057 1 3B-001 SBL3001 Steven B. Lionel 4-Oct-1982

```

RENAME
V04-000

VAX/VMS RENAME Utility

M 9
16-Sep-1984 00:25:56 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:08:48 [CLIUTL.SRC]RENAME.B32;1

Page 2
(1)

:	58	0058	1	!
:	59	0059	1	!
:	60	0060	1	!
:	61	0061	1	!--
:	62	0062	1	!

Rewritten from RENAME V03-000 to use new CLI interface,
LIB\$RENAME_FILE, and file selection qualification routines.


```

64 0063 1 %SBTTL 'Declarations'
65 0064 1
66 0065 1 SWITCHES:
67 0066 1
68 0067 1
69 0068 1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
70 0069 1
71 0070 1
72 0071 1 LINKAGES:
73 0072 1
74 0073 1 NONE
75 0074 1
76 0075 1 TABLE OF CONTENTS:
77 0076 1
78 0077 1
79 0078 1 FORWARD ROUTINE
80 0079 1 RENAME_MAIN,
81 0080 1 CONFIRM_ROUTINE,
82 0081 1 LOG_ROUTINE: NOVALUE,
83 0082 1 ERROR_ROUTINE,
84 0083 1 ERROR_HANDLER;
85 0084 1
86 0085 1
87 0086 1 INCLUDE FILES:
88 0087 1
89 0088 1
90 0089 1 LIBRARY 'SYS$LIBRARY:STARLET';
91 0090 1
92 0091 1
93 0092 1 MACROS:
94 0093 1
95 0094 1 NONE
96 0095 1
97 0096 1 EQUATED SYMBOLS:
98 0097 1
99 0098 1
100 0099 1 +
101 0100 1 Define shared messages needed by RENAME.
102 0101 1 -
103 0102 1
104 P 0103 1 $$SHR MSGDEF (MSG, 144, LOCAL,
105 P 0104 1 (NOFILES, INFO), No files selected
106 P 0105 1 (OPENIN, ERROR), Error opening input file
107 P 0106 1 (OPENOUT, ERROR), Error opening output file
108 P 0107 1 (RENAMED, INFO), File renamed
109 P 0108 1 (SEARCHFAIL, ERROR) ! Could not find file
110 0109 1 );
111 0110 1
112 0111 1
113 0112 1 FIELDS:
114 0113 1
115 0114 1
116 0115 1 +
117 0116 1 Data structure LOCAL_BLOCK (defined by LB_FIELDS) is passed as the
118 0117 1 "user_argument" to LIB$RENAME_FILE, and thus to each of the action
119 0118 1 routines.
120 0119 1 -

```

```

! Main routine
! Confirm file to be renamed
! Log successful renames
! Report errors
! Intercept signalled errors

```

```
! System symbols
```

```

121 0120 1
122 0121 1 FIELD
123 0122 1 LB_FIELDS =
124 0123 1 SET
125 0124 1
126 0125 1 LB_L_WORST_ERROR = [0,0,32,0], ! Worst error so far
127 0126 1 LB_L_QUAL_CONTEXT = [4,0,32,0], ! Context for LIB$QUAL_XXX
128 0127 1 LB_L_FLAGS = [8,0,32,0], ! Flags
129 0128 1 LB_V_LOG = [8,0,1,0], ! Set if /LOG specified
130 0129 1 LB_V_FILE_SELECTED = [8,1,1,0] ! At least one file was renamed
131 0130 1
132 0131 1 TES;
133 0132 1 LITERAL
134 0133 1 LB_K_SIZE = 12; ! Size of LOCAL_BLOCK in bytes
135 0134 1
136 0135 1
137 0136 1 PSECTS:
138 0137 1
139 0138 1
140 0139 1 PSECT
141 0140 1 CODE = $CODE (READ, NOWRITE, EXECUTE, SHARE, PIC,
142 0141 1 ADDRESSING_MODE (WORD_RELATIVE)),
143 0142 1 PLIT = $CODE (READ, NOWRITE, EXECUTE, SHARE, PIC,
144 0143 1 ADDRESSING_MODE (WORD_RELATIVE)),
145 0144 1 OWN = $DATA (READ, WRITE, NOEXECUTE, NOSHARE, PIC,
146 0145 1 ADDRESSING_MODE (LONG_RELATIVE)),
147 0146 1 GLOBAL = $DATA (READ, WRITE, NOEXECUTE, NOSHARE, PIC,
148 0147 1 ADDRESSING_MODE (LONG_RELATIVE));
149 0148 1
150 0149 1
151 0150 1 OWN STORAGE:
152 0151 1
153 0152 1 NONE
154 0153 1
155 0154 1 EXTERNAL REFERENCES:
156 0155 1
157 0156 1
158 0157 1 EXTERNAL ROUTINE
159 0158 1 CLISGET VALUE, ! Get value from command line
160 0159 1 CLISPRESENT, ! Is qualifier present?
161 0160 1 LIB$RENAME FILE, ! Rename files
162 0161 1 LIB$QUAL_FILE_MATCH, ! Match selection qualifiers
163 0162 1 LIB$QUAL_FILE_PARSE; ! Parse selection qualifiers
164 0163 1
165 0164 1 EXTERNAL LITERAL
166 0165 1 LIB$FILFAIMAT, ! File failed to match qualifiers
167 0166 1 LIB$QUICONACT, ! Quit confirm action
168 0167 1 LIB$QUIPRO, ! Quit processing
169 0168 1 RENAMES_NOTRENAMED, ! File not renamed
170 0169 1 RENAMES_NOTSAMEDEV; ! Can't rename across devices
171 0170 1

```



```

173 0171 1 %SBTTL 'RENAME_MAIN - Main routine'
174 0172 1 ROUTINE RENAME_MAIN =
175 0173 1
176 0174 1 !++
177 0175 1 FUNCTIONAL DESCRIPTION:
178 0176 1
179 0177 1 This is the main routine for RENAME. It calls LIB$QUAL_FILE_PARSE
180 0178 1 to parse the file selection qualifiers, sets up the local storage
181 0179 1 block, and calls LIB$RENAME_FILE to do the work. LIB$RENAME_FILE
182 0180 1 will call other routines in this module as action routines to
183 0181 1 process the command. Upon return from LIB$RENAME_FILE, the most
184 0182 1 severe error status will be returned as the exit status of RENAME
185 0183 1
186 0184 1 CALLING SEQUENCE:
187 0185 1
188 0186 1 ret_status.wlc.v = RENAME ()
189 0187 1
190 0188 1 FORMAL PARAMETERS:
191 0189 1
192 0190 1 NONE
193 0191 1
194 0192 1 IMPLICIT INPUTS:
195 0193 1
196 0194 1 Command line
197 0195 1
198 0196 1 IMPLICIT OUTPUTS:
199 0197 1
200 0198 1 NONE
201 0199 1
202 0200 1 COMPLETION STATUS:
203 0201 1
204 0202 1 SSS_NORMAL Normal successful completion
205 0203 1 RENAMES_NOFILES No files selected
206 0204 1 Other Worst error encountered
207 0205 1
208 0206 1 SIDE EFFECTS:
209 0207 1
210 0208 1 Renames one or more files.
211 0209 1
212 0210 1 !--
213 0211 1
214 0212 2 BEGIN
215 0213 2
216 0214 2 LOCAL
217 0215 2 LOCAL_BLOCK: VOLATILE BLOCK [LB_K_SIZE, BYTE] FIELD (LB_FIELDS),
218 0216 2 QUAL_FLAGS, Qualifier select flags
219 0217 2 RENAME_FLAGS, Flags for LIB$RENAME_FILE
220 0218 2 OLD_NAME_DSC: BLOCK [8, BYTE], Old name descriptor
221 0219 2 NEW_NAME_DSC: BLOCK [8, BYTE], New name descriptor
222 0220 2 OLD_RSN_DSC: BLOCK [8, BYTE], Old RSN descriptor
223 0221 2 NEW_RSN_DSC: BLOCK [8, BYTE], New RSN descriptor
224 0222 2 CLI_STRING_DSC: BLOCK [8, BYTE], Qualifier and parameter names
225 0223 2 FILESCAN_CONTEXT, Related file context pointer
226 0224 2 STATUS1; Return status
227 0225 2
228 0226 2 !+
229 0227 2 ! Enable error handler which will keep track of the worst error

```

```

230      0228      | encountered so far. It will also unwind to the caller of RENAME
231      0229      | (the CLI) if the status LIB$QUIPRO is signalled. Note that because
232      0230      | LOCAL_BLOCK is a VOLATILE enable argument, BLISS zeroes it for us.
233      0231      |
234      0232      |
235      0233      |
236      0234      |
237      0235      |
238      0236      |
239      0237      | +
240      0238      | Set up strings and descriptors. All strings are dynamic.
241      0239      |
242      0240      |
243      0241      | FILESCAN_CONTEXT = 0;
244      0242      | OLD_RSN_DSC [DSC$W_LENGTH] = 0;
245      0243      | OLD_RSN_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
246      0244      | OLD_RSN_DSC [DSC$B_CLASS] = DSC$K_CLASS_D;
247      0245      | OLD_RSN_DSC [DSC$A_POINTER] = 0;
248      0246      | NEW_RSN_DSC [DSC$W_LENGTH] = 0;
249      0247      | NEW_RSN_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
250      0248      | NEW_RSN_DSC [DSC$B_CLASS] = DSC$K_CLASS_D;
251      0249      | NEW_RSN_DSC [DSC$A_POINTER] = 0;
252      0250      | OLD_NAME_DSC [DSC$W_LENGTH] = 0;           | Initially zero length
253      0251      | OLD_NAME_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;   | Text
254      0252      | OLD_NAME_DSC [DSC$B_CLASS] = DSC$K_CLASS_D; | Dynamic
255      0253      | OLD_NAME_DSC [DSC$A_POINTER] = 0;         | No allocation
256      0254      | NEW_NAME_DSC [DSC$W_LENGTH] = 0;       | Initially zero length
257      0255      | NEW_NAME_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;   | Text
258      0256      | NEW_NAME_DSC [DSC$B_CLASS] = DSC$K_CLASS_D; | Dynamic
259      0257      | NEW_NAME_DSC [DSC$A_POINTER] = 0;         | No allocation
260      0258      | CLI_STRING_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
261      0259      | CLI_STRING_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
262      0260      |
263      0261      | +
264      0262      | Indicate which qualifiers we want parsed.
265      0263      |
266      0264      |
267      0265      |
268      0266      |
269      0267      |
270      0268      |
271      0269      |
272      0270      |
273      0271      |
274      0272      |
275      0273      |
276      0274      | +
277      0275      | Parse file selection qualifiers and save context.
278      0276      |
279      0277      |
280      0278      | LIB$QUAL_FILE_PARSE (QUAL_FLAGS, LOCAL_BLOCK [LB_L_QUAL_CONTEXT]);
281      0279      |
282      0280      |
283      0281      | +
284      0282      | See if /LOG was specified.
285      0283      |
286      0284      |

```



```

287 0285 CLI_STRING_DSC [DSC$A_POINTER] = UPLIT BYTE ('LOG');
288 0286 IF CLISPRESNT (CLI_STRING_DSC)
289 0287 THEN
290 0288     LOCAL_BLOCK [LB_V_LOG] = 1;
291 0289
292 0290
293 0291 !+ See if /NONEW_VERSION was specified.
294 0292 !-
295 0293
296 0294 CLI_STRING_DSC [DSC$W_LENGTH] = %CHARCOUNT ('NEW_VERSION');
297 0295 CLI_STRING_DSC [DSC$A_POINTER] = UPLIT BYTE ('NEW_VERSION');
298 0296 RENAME_FLAGS = 0;
299 0297 IF NOT CLISPRESNT (CLI_STRING_DSC)
300 0298 THEN
301 0299     RENAME_FLAGS = 1;      ! SET V_NONEW_VERSION bit
302 0300
303 0301 !+
304 0302 !- Get new filename, since there can only be one.
305 0303 !-
306 0304
307 0305 CLI_STRING_DSC [DSC$W_LENGTH] = %CHARCOUNT ('P2');
308 0306 CLI_STRING_DSC [DSC$A_POINTER] = UPLIT BYTE ('P2');
309 0307 CLISGET_VALUE (CLI_STRING_DSC, NEW_NAME_DSC);
310 0308
311 0309 !+
312 0310 !- Until all "old" filespecs are processed, get the next old filespec
313 0311 !- and call LIB$RENAME_FILE for each one. Use the previous old resultant
314 0312 !- filename as the related filename for each iteration.
315 0313 !-
316 0314
317 0315 CLI_STRING_DSC [DSC$W_LENGTH] = %CHARCOUNT ('P1');
318 0316 CLI_STRING_DSC [DSC$A_POINTER] = UPLIT BYTE ('P1');
319 0317 WHILE (CLISGET_VALUE (CLI_STRING_DSC, OLD_NAME_DSC)) DO
320 0318 BEGIN
321 0319     STATUS1 = LIB$RENAME_FILE (
322 0320         OLD_NAME_DSC,      ! Old name
323 0321         NEW_NAME_DSC,     ! New name
324 0322         0,                ! No default name
325 0323         0,                ! Related name - previous old RSN
326 0324         RENAME_FLAGS,    ! Rename flags
327 0325         LOG_ROUTINE,     ! Log routine
328 0326         ERROR_ROUTINE,  ! Error routine
329 0327         CONFIRM_ROUTINE, ! Confirm routine
330 0328         LOCAL_BLOCK,     ! User argument
331 0329         OLD_RSN_DSC,     ! Old resultant name
332 0330         NEW_RSN_DSC,     ! New resultant name
333 0331         FILE$CAR_CONTEXT); ! Related file context
334 0332
335 0333 IF NOT .STATUS1
336 0334 THEN
337 0335     SIGNAL_STOP (.STATUS1);
338 0336 END;
339 0337
340 0338 !+
341 0339 !- If an error occurred, return the worst error.
342 0340 !- If no files were selected, return an informational to that effect.
343 0341 !-

```

RENAME
V04-000

VAX/VMS RENAME Utility
RENAME_MAIN - Main routine

F 10
16-Sep-1984 00:25:56
14-Sep-1984 12:08:48

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]RENAME.B32;1

Page 8
(3)

```

344 0342 2
345 0343 IF .LOCAL_BLOCK [LB_L_WORST_ERROR] NEQ 0
346 0344 THEN
347 0345 RETURN .LOCAL_BLOCK [LB_L_WORST_ERROR];
348 0346
349 0347 IF NOT .LOCAL_BLOCK [LB_V_FILE_SELECTED]
350 0348 THEN
351 0349 BEGIN
352 0350 SIGNAL (MSG$_NOFILES); ! Display message 'no files selected'
353 0351 RETURN MSG$_NOFILES; ! Won't get displayed again
354 0352 END;
355 0353
356 0354 RETURN SS$_NORMAL;
357 0355
358 0356 END;
```

! End of routine RENAME_MAIN

.TITLE RENAME VAX/VMS RENAME Utility
.IDENT \V04-000\

.PSECT \$CODE,NOWRT, SHR, PIC,2

```

4E 4F 49 53 52 45 56 5F 47 4F 4C 00000 P.AAA: .ASCII \LOG\
45 4E 00003 P.AAB: .ASCII \NEW_VERSION\
32 50 0000E P.AAC: .ASCII \P2\
31 50 00010 P.AAD: .ASCII \P1\
```

.EXTRN CLISGET VALUE, CLISPRESNT
.EXTRN LIB\$RENAME_FILE
.EXTRN LIB\$QUAL_FILE_MATCH
.EXTRN LIB\$QUAL_FILE_PARSE
.EXTRN LIB\$FILFIMAT, LIB\$QUICONACT
.EXTRN LIB\$QUIPRO, RENAMES\$_NOTRENAMED
.EXTRN RENAMES\$_NOTSAMEDEV

001C 00000 RENAME_MAIN:

```

54 00000000G 00 9E 00002 .WORD Save R2,R3,R4
53 00000000G 00 9E 00009 MOVAB CLISGET VALUE, R4
5E 38 C2 00010 MOVAB CLISPRESNT, R3
2C AE 7C 00013 SUBL2 #56, SP
34 AE D4 00016 CLRQ LOCAL_BLOCK
6D 0103 CF DE 00019 CLRL LOCAL_BLOCK
7E D4 0001E MOVAL 78, (FP)
18 AE 020E0000 8F D0 00020 CLRL FILESCAN CONTEXT
1C AE D4 00028 MOVL #34471936, OLD_RSN_DSC
10 AE 020E0000 8F D0 0002B CLRL OLD_RSN_DSC+4
14 AE D4 00033 MOVL #34471936, NEW_RSN_DSC
28 AE 020E0000 8F D0 00036 CLRL NEW_RSN_DSC+4
2C AE D4 0003E MOVL #34471936, OLD_NAME_DSC
20 AE 020E0000 8F D0 00041 CLRL OLD_NAME_DSC+4
24 AE D4 00049 MOVL #34471936, NEW_NAME_DSC
0A AE 010E 8F B0 0004C CLRL NEW_NAME_DSC+4
7E 01FF 8F 3C 00052 MOVW #270, CLT_STRING_DSC+2
38 AE 9F 00057 MOVZWL #511, QUAL_FLAGS
04 AE 9F 0005A PUSHAB LOCAL_BLOCK+4
00000000G 00 02 FB 0005D PUSHAB QUAL_FLAGS
CALLS #2, LIB$QUAL_FILE_PARSE
```

0172
0212
0240
0241
0244
0245
0248
0249
0252
0253
0256
0257
0271
0278

RENAME
V04-000

VAX/VMS RENAME Utility
RENAME_MAIN - Main routine

G 10

16-Sep-1984 00:25:56

14-Sep-1984 12:08:48

VAX-11 B11sg-32 V4.0-742

[CLIUTL.SRC]RENAME.B32:1

Page 9

(3)

0C	AE		03	80	00064	MOVW	#3, CLI_STRING_DSC	0284
10	AE	83	AF	9E	00068	MOVAB	P.AAA, CLI_STRING_DSC+4	0285
		0C	AE	9F	0006D	PUSHAB	CLI_STRING_DSC	0286
	63		01	FB	00070	CALLS	#1, CLISPRESNT	
	04		50	E9	00073	BLBC	R0, 18	
3C	AE		01	88	00076	BISB2	#1, LOCAL_BLOCK+8	0288
0C	AE		0B	80	0007A	MOVW	#11, CLI_STRING_DSC	0294
10	AE	FF6F	CF	9E	0007E	MOVAB	P.AAB, CLI_STRING_DSC+4	0295
		08	AE	D4	00084	CLRL	RENAME_FLAGS	0296
		0C	AE	9F	00087	PUSHAB	CLI_STRING_DSC	0297
	63		01	FB	0008A	CALLS	#1, CLISPRESNT	
	04		50	E8	0008D	BLBS	R0, 28	
08	AE		01	D0	00090	MOVL	#1, RENAME_FLAGS	0299
0C	AE		02	80	00094	MOVW	#2, CLI_STRING_DSC	0305
10	AE	FF60	CF	9E	00098	MOVAB	P.AAC, CLI_STRING_DSC+4	0306
		24	AE	9F	0009E	PUSHAB	NEW_NAME_DSC	0307
		10	AE	9F	000A1	PUSHAB	CLI_STRING_DSC	
	64		02	FB	000A4	CALLS	#2, CLISGET_VALUE	
0C	AE		02	80	000A7	MOVW	#2, CLI_STRING_DSC	0315
10	AE	FF4F	CF	9E	000AB	MOVAB	P.AAD, CLI_STRING_DSC+4	0316
		2C	AE	9F	000B1	PUSHAB	OLD_NAME_DSC	0317
		10	AE	9F	000B4	PUSHAB	CLI_STRING_DSC	
	64		02	FB	000B7	CALLS	#2, CLISGET_VALUE	
	3B		50	E9	000BA	BLBC	R0, 48	
		04	AE	9F	000BD	PUSHAB	FILESCAN_CONTEXT	0319
		18	AE	9F	000C0	PUSHAB	NEW_RSN_DSC	
		24	AE	9F	000C3	PUSHAB	OLD_RSN_DSC	
		40	AE	9F	000C6	PUSHAB	LOCAL_BLOCK	
	0000V		CF	9F	000C9	PUSHAB	CONFIRM_ROUTINE	
	0000V		CF	9F	000CD	PUSHAB	ERROR_ROUTINE	
	0000V		CF	9F	000D1	PUSHAB	LOG_ROUTINE	
		24	AE	9F	000D5	PUSHAB	RENAME_FLAGS	
			7E	7C	000D8	CLRG	-(SP)	
		4C	AE	9F	000DA	PUSHAB	NEW_NAME_DSC	
		58	AE	9F	000DD	PUSHAB	OLD_NAME_DSC	
00000000G	00		0C	FB	000E0	CALLS	#12, LIB\$RENAME_FILE	
	52		50	D0	000E7	MOVL	R0, STATUS1	
	C4		52	E8	000EA	BLBS	STATUS1, 38	0333
			52	DD	000ED	PUSHL	STATUS1	0335
00000000G	00		01	FB	000EF	CALLS	#1, LIB\$STOP	
			B9	11	000F6	BRB	38	0317
		34	AE	D5	000F8	TSTL	LOCAL_BLOCK	0343
			05	13	000FB	BEQL	58	
	50	34	AE	D0	000FD	MOVL	LOCAL_BLOCK, R0	0345
			04	04	00101	RET		
15	3C	AE	01	E0	00102	BBS	#1, LOCAL_BLOCK+8, 68	0347
		00901333	8F	DD	00107	PUSHL	#9442099	0350
00000000G	00		01	FB	0010D	CALLS	#1, LIB\$SIGNAL	
	50	00901333	8F	D0	00114	MOVL	#9442099, R0	0351
			04	04	0011B	RET		
	50		01	D0	0011C	MOVL	#1, R0	0354
			04	04	0011F	RET		0356
			0000	00	00120	.WORD	Save nothing	0212
	50		AC	D0	00122	MOVL	8(AP), R0	
	50		A0	D0	00126	MOVL	4(R0), R0	
		F4	A0	9F	0012A	PUSHAB	LOCAL_BLOCK	
			01	DD	0012D	PUSHL	#1	

RENAME
V04-000

VAX/VMS RENAME Utility
RENAME_MAIN - Main routine

H 10
16-Sep-1984 00:25:56 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:08:48 [CLIUTL.SRC]RENAME.B32;1

Page 10
(3)

0000V	7E	04	5E	DD	0012F	PUSHL	SP
	CF		AC	7D	00131	MOVQ	4(AP), -(SP)
			03	FB	00135	CALLS	#3, ERROR_HANDLER
			04	0013A	RET		

; Routine Size: 315 bytes, Routine Base: \$CODE + 0012

```

360 0357 1 %SBYTL 'CONFIRM_ROUTINE - Confirm that rename should be done'
361 0358 1 ROUTINE CONFIRM_ROUTINE (
362 0359 1     OLD_RSN: REF BLOCK [, BYTE],      ! Old resultant string descriptor
363 0360 1     NEW_RSN: REF BLOCK [, BYTE],      ! New resultant string descriptor
364 0361 1     FAB: REF BLOCK [, BYTE],         ! File's FAB
365 0362 1     LOCAL_BLOCK: REF BLOCK (LB_K_SIZE, BYTE) FIELD (LB_FIELDS)
366 0363 1 ) =
367 0364 1
368 0365 1
369 0366 1
370 0367 1
371 0368 1
372 0369 1
373 0370 1
374 0371 1
375 0372 1
376 0373 1
377 0374 1
378 0375 1
379 0376 1
380 0377 1
381 0378 1
382 0379 1
383 0380 1
384 0381 1
385 0382 1
386 0383 1
387 0384 1
388 0385 1
389 0386 1
390 0387 1
391 0388 1
392 0389 1
393 0390 1
394 0391 1
395 0392 1
396 0393 1
397 0394 1
398 0395 1
399 0396 1
400 0397 1
401 0398 1
402 0399 1
403 0400 1
404 0401 1
405 0402 1
406 0403 1
407 0404 1
408 0405 1
409 0406 1
410 0407 1
411 0408 1
412 0409 1
413 0410 1
414 0411 1
415 0412 1
416 0413 1

!+
FUNCTIONAL DESCRIPTION:

    This routine is called by LIBSRENAME_FILE to confirm that the
    file described by FAB should be renamed.

CALLING SEQUENCE:

    confirm.wv.v = CONFIRM_ROUTINE (OLD_RSN.rt.dd, NEW_RSN.rt.dd,
    FAB.r.r.r, LOCAL_BLOCK.mr.r)

FORMAL PARAMETERS:

    OLD_RSN      - Old resultant name.
    NEW_RSN      - New resultant name.
    FAB          - FAB that describes file to be renamed.
    LOCAL_BLOCK  - Block of local information.

IMPLICIT INPUTS:

    NONE

IMPLICIT OUTPUTS:

    NONE

ROUTINE VALUE:

    TRUE if file should be renamed
    FALSE if file should not be renamed

SIDE EFFECTS:

    NONE

--

BEGIN

LOCAL

    PROMPT_ARGS: VECTOR [2, LONG],      ! FAO argument list for prompt
    PROMPT_DSC: BLOCK [8, BYTE],        ! Descriptor for prompt string
    MATCH_STATUS;                        ! Status from QUAL_FILE_MATCH

!+
! Set up PROMPT_ARGS and PROMPT_DSC.

```

```

417 0414 2  !-
418 0415
419 0416
420 0417 PROMPT_ARGS [0] = OLD_RSN [0,0,0,0];      ! Descriptor address
421 0418 PROMPT_ARGS [1] = NEW_RSN [0,0,0,0];      ! Descriptor address
422 0419 PROMPT_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
423 0420 PROMPT_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
424 0421 PROMPT_DSC [DSC$W_LENGTH] = %CHARCOUNT ('!AS --> !AS ? (Y,N,Q,ALL): ');
425 0422 PROMPT_DSC [DSC$A_POINTER] = UPLIT BYTE ('!AS --> !AS ? (Y,N,Q,ALL): ');
426 0423
427 0424 !+
428 0425 ! Call LIB$QUAL_FILE_MATCH to see if this file meets the criteria.
429 0426 !-
430 0427 MATCH_STATUS = LIB$QUAL_FILE_MATCH (
431 0428     LOCAL_BLOCK [LB_L_QUAL_CONTEXT],
432 0429     FAB [0,0,0,0],
433 0430     0,      ! File name omitted
434 0431     PROMPT_DSC,
435 0432     PROMPT_ARGS,
436 0433     0);    ! Prompt routine omitted
437 0434
438 0435 !+
439 0436 ! If a failure status other than "failed to match" was returned,
440 0437 ! signal it. If it is LIB$QUIPRO, ERROR_HANDLER will unwind back to
441 0438 ! the CLI with a success status.
442 0439 !-
443 0440
444 0441 IF NOT .MATCH_STATUS
445 0442 THEN
446 0443     IF .MATCH_STATUS NEQU LIB$FILFAMAT
447 0444     THEN
448 0445         IF (.MATCH_STATUS EQLU LIB$QUIPRO) OR (.FAB [FAB$L_STS])
449 0446         THEN
450 0447             SIGNAL (.MATCH_STATUS)
451 0448         ELSE
452 0449             BEGIN
453 0450                 !+
454 0451                 ! LIB$QUAL_FILE_MATCH couldn't open the file.
455 0452                 ! Signal "error opening file".
456 0453                 !-
457 0454
458 0455                 LOCAL
459 0456                     NAM: REF BLOCK [, BYTE],
460 0457                     RSN_DESCR: BLOCK [8, BYTE];
461 0458                 NAM = .FAB [FAB$L_NAM];
462 0459                 RSN_DESCR [DSC$B_DTYPE] = DSC$K_DTYPE_T;
463 0460                 RSN_DESCR [DSC$B_CLASS] = DSC$K_CLASS_S;
464 0461                 RSN_DESCR [DSC$W_LENGTH] = .NAM [NAM$B_RSL];
465 0462                 RSN_DESCR [DSC$A_POINTER] = .NAM [NAM$B_RSA];
466 0463                 SIGNAL (MSG$OPERRIN, 1, RSN_DESCR,
467 0464                     .FAB [FAB$L_STS], .FAB [FAB$L_STV]);
468 0465                 END;
469 0466
470 0467 !+
471 0468 ! If this file was selected, indicate that in LOCAL_BLOCK.
472 0469 !-
473 0470

```


RENAME
V04-000

VAX/VMS RENAME Utility

CONFIRM_ROUTINE - Confirm that rename should be

K 10

16-Sep-1984 00:25:56

14-Sep-1984 12:08:48

VAX-11 Bliss-32 V4.0-742

[CLIUTL.SRC]RENAME.B32:1

Page 13

(4)

```

474 0471 2 IF .MATCH_STATUS
475 0472 THEN
476 0473 LOCAL_BLOCK [LB_V_FILE_SELECTED] = 1;
477 0474
478 0475
479 0476
480 0477
481 0478
482 0479
483 0480
484 0481 1

```

Return to LIB\$RENAME_FILE with match status.

RETURN .MATCH_STATUS;

END;

! End of routine CONFIRM_ROUTINE

28 20 3F 20 53 41 21 20 3E 2D 2D 20 53 41 21 0014D P.AAE: .ASCII \!AS --> !AS ? (Y,N,Q,ALL): \

003C 00000 CONFIRM_ROUTINE:

55	00000000G	00	9E	00002	.WORD	Save R2,R3,R4,R5	0358
5E		18	C2	00009	MOVAB	LIB\$SIGNAL, R5	
10	AE	04	AC	7D	SUBL2	#24, SP	0416
08	AE	010E001B	8F	D0	MOVQ	OLD_RSN, PROMPT_ARGS	0420
0C	AE	C9	AF	9E	MOVL	#17894747, PROMPT_DSC	0421
			7E	D4	MOVAB	P.AAE, PROMPT_DSC+4	0429
		14	AE	9F	CLRL	-(SP)	
		10	AE	9F	PUSHAB	PROMPT_ARGS	
			7E	D4	PUSHAB	PROMPT_DSC	
52	0C	AC	7D	00028	CLRL	-(SP)	
		52	DD	0002C	MOVQ	FAB, R2	
		04	A3	9F	PUSHL	R2	
00000000G	00	06	FB	00031	PUSHAB	4(R3)	0428
54		50	D0	00038	CALLS	#6, LIB\$QUAL_FILE_MATCH	0429
45		54	E8	0003B	MOVL	R0, MATCH_STATUS	
00000000G	8F	54	D1	0003E	BLBS	MATCH_STATUS, 4\$	0441
		39	13	00045	CMPL	MATCH_STATUS, #LIB\$_FILFAMAT	0443
00000000G	8F	54	D1	00047	BEQL	3\$	
		04	13	0004E	CMPL	MATCH_STATUS, #LIB\$_QUIPRO	0445
	07	08	A2	E9	BEQL	1\$	
			54	DD	BLBC	8(R2), 2\$	
65			01	FB	PUSHL	MATCH_STATUS	0447
			25	11	CALLS	#1, LIB\$SIGNAL	
50	28		A2	D0	BRB	3\$	
02	AE	010E	8F	B0	MOVL	40(R2), NAM	0458
6E	03		A0	9B	MOVW	#270, RSN_DESCR+2	0459
04	AE	04	A0	D0	MOVZBW	3(NAM), RSN_DESCR	0461
7E	08		A2	7D	MOVL	4(NAM), RSN_DESCR+4	0462
	08		AE	9F	MOVQ	8(R2), -(SP)	0464
			01	DD	PUSHAB	RSN_DESCR	0463
	0090109A		8F	DD	PUSHL	#1	
65			05	FB	PUSHL	#9441434	
04			54	E9	CALLS	#5, LIB\$SIGNAL	0471
08	A3		02	88	BLBC	MATCH_STATUS, 5\$	0473
50			54	D0	BISB2	#2, 8TR3)	0479
			04	0008A	MOVL	MATCH_STATUS, R0	0481
					RET		

RENAME
V04-000

VAX/VMS RENAME Utility
CONFIRM_ROUTINE - Confirm that rename should be

L 10
16-Sep-1984 00:25:56
14-Sep-1984 12:08:48

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]RENAME.B32;1

Page 14
(4)

; Routine Size: 139 bytes, Routine Base: \$CODE + 0168

```

486 0482 1 $SBTTL 'LOG_ROUTINE - Log successful rename'
487 0483 1 ROUTINE LOG_ROUTINE (
488 0484 1     OLD_RSN: REF BLOCK [, BYTE],           ! Old resultant name
489 0485 1     NEW_RSN: REF BLOCK [, BYTE],         ! New resultant name
490 0486 1     LOCAL_BLOCK: REF BLOCK [LB_K_SIZE, BYTE] FIELD (LB_FIELDS)
491 0487 1 ): NOVALUE =
492 0488 1
493 0489 1 --
494 0490 1 **
495 0491 1 FUNCTIONAL DESCRIPTION:
496 0492 1     This routine is called by LIB$RENAME_FILE after a successful
497 0493 1     rename.  If /LOG was specified, a message indicating the names
498 0494 1     of the files renamed is displayed.
499 0495 1
500 0496 1 CALLING SEQUENCE:
501 0497 1     CALL LOG_ROUTINE (OLD_RSN.rt.dd, NEW_RSN.rt.dd,
502 0498 1                     LOCAL_BLOCK.mr.r)
503 0499 1
504 0500 1 FORMAL PARAMETERS:
505 0501 1
506 0502 1     OLD_RSN           - Old resultant name.
507 0503 1
508 0504 1     NEW_RSN           - New resultant name.
509 0505 1
510 0506 1     LOCAL_BLOCK       - Block of local information.
511 0507 1
512 0508 1 IMPLICIT INPUTS:
513 0509 1
514 0510 1     NONE
515 0511 1
516 0512 1 IMPLICIT OUTPUTS:
517 0513 1
518 0514 1     NONE
519 0515 1
520 0516 1 ROUTINE VALUE:
521 0517 1
522 0518 1     NONE
523 0519 1
524 0520 1 SIDE EFFECTS:
525 0521 1
526 0522 1     NONE
527 0523 1
528 0524 1 --
529 0525 1
530 0526 1 BEGIN
531 0527 1
532 0528 1     !+
533 0529 1     ! If /LOG was specified, log the rename.
534 0530 1     !-
535 0531 1
536 0532 1     IF .LOCAL_BLOCK [LB_V_LOG]
537 0533 1     THEN
538 0534 1         SIGNAL (MSG$_RENAMED, 2, OLD_RSN [0,0,0,0], NEW_RSN [0,0,0,0]);
539 0535 1
540 0536 1 RETURN;
541 0537 1
542 0538 1

```


RENAME
V04-000

VAX/VMS RENAME Utility
LOG_ROUTINE - Log successful rename

N 10
16-Sep-1984 00:25:56
14-Sep-1984 12:08:48

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]RENAME.B32;1

Page 16
(5)

; 543

0539 1 END;

! End of routine LOG_ROUTINE

```
0000 00000 LOG_ROUTINE:
          50      0C      AC      D0 00002      .WORD      Save nothing
          13      08      A0      E9 00006      MOVL      LOCAL_BLOCK, R0
          7E      04      AC      7D 0000A      BLBC      8(R0), 1$
          02      DD 0000E      MOVQ      OLD_RSN, -(SP)
00000000G 00 009012A3 8F      DD 00010      PUSHL     #2
          04      FB 00016      PUSHL     #9441955
          04 0001D 1$:      CALLS     #4, LIB$SIGNAL
          RET
```

: 0483
: 0533
: 0535
: 0539

; Routine Size: 30 bytes. Routine Base: \$CODE + 01F3

```

545 0540 1 %SBTTL 'ERROR_ROUTINE - Report failed rename'
546 0541 1 ROUTINE ERROR_ROUTINE (
547 0542 1     OLD_RSN: REF BLOCK [, BYTE],      ! Old resultant name
548 0543 1     NEW_RSN: REF BLOCK [, BYTE],    ! New resultant name
549 0544 1     RMS_STS: REF BLOCK [, BYTE],    ! RMS STS status code
550 0545 1     RMS_STV: REF BLOCK [, BYTE],    ! RMS STV status code
551 0546 1     ERROR_SOURCE: REF VECTOR [, LONG], ! Source of error code
552 0547 1     LOCAL_BLOCK: REF BLOCK [LB_K_SIZE, BYTE] FIELD (LB_FIELDS)
553 0548 1 ) =
554 0549 1
555 0550 1 ++
556 0551 1 FUNCTIONAL DESCRIPTION:
557 0552 1
558 0553 1     This routine is called by LIB$RENAME FILE after a failed
559 0554 1     rename. The error message is signalled.
560 0555 1
561 0556 1 CALLING SEQUENCE:
562 0557 1
563 0558 1     status.wlc.v = ERROR_ROUTINE (OLD_RSN.rt.dd, NEW_RSN.rt.dd,
564 0559 1     RMS_STS.rlc.r, RMS_STV.rlc.r,
565 0560 1     ERROR_SOURCE.rl.r, LOCAL_BLOCK.mr.r)
566 0561 1
567 0562 1 FORMAL PARAMETERS:
568 0563 1
569 0564 1     OLD_RSN          - Old resultant name.
570 0565 1
571 0566 1     NEW_RSN          - New resultant name.
572 0567 1
573 0568 1     RMS_STS          - The RMS STS condition code
574 0569 1
575 0570 1     RMS_STV          - The RMS STV condition code
576 0571 1
577 0572 1     ERROR_SOURCE     - An integer that indicates where the error occurred.
578 0573 1                     0 = Error from LIB$FILE_SCAN
579 0574 1                     1 = Error from $PARSE
580 0575 1                     2 = Error from $RENAME
581 0576 1     LOCAL_BLOCK      - Block of local information.
582 0577 1
583 0578 1 IMPLICIT INPUTS:
584 0579 1
585 0580 1     NONE
586 0581 1
587 0582 1 IMPLICIT OUTPUTS:
588 0583 1
589 0584 1     NONE
590 0585 1
591 0586 1 ROUTINE VALUE:
592 0587 1
593 0588 1     $$$_NORMAL
594 0589 1
595 0590 1 SIDE EFFECTS:
596 0591 1
597 0592 1     Signals the error
598 0593 1
599 0594 1 --
600 0595 1
601 0596 2 BEGIN

```

0541
0602
0608
0607

RENAME
V04-000

VAX/VMS RENAME Utility
ERROR_ROUTINE - Report failed rename

D 11
16-Sep-1984 00:25:56
14-Sep-1984 12:08:48

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]RENAME.B32;1

Page 19
(6)

		44	11	00025	BRB	6\$		
	10	BC	DD	00027	PUSHL	@RMS-STV	0614	
	0C	BC	DD	0002A	PUSHL	@RMS-ST\$		
	08	AC	DD	0002D	PUSHL	NEW_RSN	0613	
		01	DD	00030	PUSHL	#1		
	009010A2	8F	DD	00032	PUSHL	#9441442		
		31	11	00038	BRB	6\$		
000184C4	8F	0C	BC	D1 0003A	4\$:	CPL	@RMS-ST\$, #99524	0624
		16	12	00042	BNEQ	5\$		
	00000000G	8F	DD	00044	PUSHL	#RENAMES_NOTSAMEDEV	0626	
	04	AC	DD	0004A	PUSHL	OLD_RSN		
		01	DD	0004D	PUSHL	#1		
	00000000G	8F	DD	0004F	PUSHL	#RENAMES_NOTRENAMED		
62		04	FB	00055	CALLS	#4, LIB\$SIGNAL		
		14	11	00058	BRB	7\$		
	10	BC	DD	0005A	5\$:	PUSHL	@RMS-STV	0630
	0C	BC	DD	0005D	PUSHL	@RMS-ST\$		
	04	AC	DD	00060	PUSHL	OLD_RSN	0629	
		01	DD	00063	PUSHL	#1		
	0090109A	8F	DD	00065	PUSHL	#9441434		
62		05	FB	0006B	6\$:	CALLS	#5, LIB\$SIGNAL	
50		01	DD	0006E	7\$:	MOVL	#1, R0	0634
		04	00071	RET			0636	

; Routine Size: 114 bytes, Routine Base: \$CODE + 0211

```

643 0637 1 XSBTTL 'ERROR_HANDLER - Condition handler'
644 0638 1 ROUTINE ERROR_HANDLER(
645 0639 1     SIGARGS: REF BLOCK [, BYTE],      ! Signal argument list
646 0640 1     MCHARGS: REF BLOCK [, BYTE],    ! Mechanism argument list
647 0641 1     ENBARGS: REF VECTOR [, LONG]  ! Enable argument list
648 0642 1 ) =
649 0643 1
650 0644 1 !++
651 0645 1 FUNCTIONAL DESCRIPTION:
652 0646 1
653 0647 1     This is the condition handler for RENAME_MAIN. It stores the
654 0648 1     'worst error' in the LOCAL_BLOCK. If the error is LIB$QUIPRO,
655 0649 1     caused by the user typing Quit to the confirm prompt, an unwind
656 0650 1     with status SS$NORMAL is done to the caller of RENAME_MAIN.
657 0651 1
658 0652 1 CALLING SEQUENCE:
659 0653 1
660 0654 1     ret_status.wlc.v = ERROR_HANDLER (sigargs.rbu.ra, mchargs.mbu.ra,
661 0655 1                                     enbargs.mbu.ra)
662 0656 1
663 0657 1 FORMAL PARAMETERS:
664 0658 1
665 0659 1     SIGARGS      - The signal arguments list
666 0660 1
667 0661 1     MCHARGS      - The mechanism arguments list
668 0662 1
669 0663 1     ENBARGS      - The enable arguments list. It is an array of
670 0664 1                   two longwords, the second of which is the
671 0665 1                   address of LOCAL_BLOCK.
672 0666 1
673 0667 1 IMPLICIT INPUTS:
674 0668 1
675 0669 1     NONE
676 0670 1
677 0671 1 IMPLICIT OUTPUTS:
678 0672 1
679 0673 1     NONE
680 0674 1
681 0675 1 ROUTINE VALUE:
682 0676 1
683 0677 1     SS$RESIGNAL
684 0678 1
685 0679 1 SIDE EFFECTS:
686 0680 1
687 0681 1     NONE
688 0682 1
689 0683 1 --
690 0684 1
691 0685 2 BEGIN
692 0686 2
693 0687 2 LOCAL
694 0688 2     LOCAL_BLOCK: REF BLOCK [, BYTE] FIELD (LB_FIELDS),
695 0689 2     OLD_CODE: BLOCK [4, BYTE],      ! Old status
696 0690 2     NEW_CODE: BLOCK [4, BYTE];      ! New status
697 0691 2
698 0692 2 !+
699 0693 2 ! Check to see if this is an unwind. If so, return.

```

```

700      0694      2      ! Also return if the message is a success status.
701      0695      !-
702      0696
703      0697      NEW_CODE = .SIGARGS [CHF$! SIG_NAME];
704      0698      IF T.NEW_CODE) OR (.NEW_CODE EQLU SSS_UNWIND)
705      0699      THEN
706      0700          RETURN SSS_RESIGNAL;
707      0701
708      0702      !+
709      0703      ! Check to see if this is LIB$QUIPRO. If so, unwind.
710      0704      !-
711      0705
712      0706      IF .NEW_CODE EQLU LIB$QUIPRO
713      0707      THEN
714      0708          BEGIN
715      0709              MCHARGS [CHF$!_MCH SAVRO] = SSS_NORMAL; ! Restore status of R0
716      0710              SUNWIND (); ! Unwind to caller of establisher
717      0711              RETURN SSS_RESIGNAL;
718      0712          END;
719      0713
720      0714      !+
721      0715      ! If this status is "worse" than the previous worst status, save it.
722      0716      !-
723      0717
724      0718      LOCAL_BLOCK = .ENBARGS [1];
725      0719      OLD_CODE = .LOCAL_BLOCK [LB_L_WORST_ERROR];
726      0720      IF .NEW_CODE [STSSV_SEVERITY] GTRU .OLD_CODE [STSSV_SEVERITY]
727      0721      THEN
728      0722          LOCAL_BLOCK [LB_L_WORST_ERROR] = .NEW_CODE + STSM_INHIB_MSG;
729      0723
730      0724      RETURN SSS_RESIGNAL;
731      0725
732      0726      1      END;

```

! End of routine ERROR_HANDLER

```

                                .EXTRN SYSSUNWIND
                                000C 00000 ERROR_HANDLER:
                                .WORD
                                50      04      AC      D0 00002      Save R2,R3
                                52      04      AO      D0 00006      SIGARGS, R0
                                43      52      EB 0000A      4(R0), NEW_CODE
                                00000920 8F      52      D1 0000D      BLBS      NEW_CODE, 2$
                                3A      13 00014      CMPL      NEW_CODE, #2336
                                00000000G 8F      52      D1 00016      BEQL      2$
                                13      12 0001D      CMPL      NEW_CODE, #LIB$QUIPRO
                                50      08      AC      D0 0001F      BNEQ      1$
                                0C      AO      01      D0 00023      MOVL      MCHARGS, R0
                                00000000G 00      7E      7C 00027      MOVL      #1, 12(R0)
                                50      0C      AC      D0 00032 1$:      CLRQ      -(SP)
                                50      04      AO      D0 00036      CALLS      #2, SYSSUNWIND
                                51      60      D0 0003A      BRB      2$
                                53      00      EF 0003D      MOVL      ENBARGS, R0
                                53      00      ED 00042      MOVL      4(R0), LOCAL_BLOCK
                                51      03      00      00      MOVL      (LOCAL_BLOCK), OLD_CODE
                                52      03      00      00      EXTZV      #0, #3, OLD_CODE, R3
                                53      03      00      00      CMPZV      #0, #3, NEW_CODE, R3

```

RENAME
V04-000

VAX/VMS RENAME Utility
ERROR_HANDLER - Condition handler

6 11
16-Sep-1984 00:25:56
14-Sep-1984 12:08:48

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]RENAME.B32;1

Page 22
(7)

60	10000000	07	1B	00047	BLEQU	2\$	
50	0918	E2	9E	00049	MOVAB	268435456(R2),	(LOCAL_BLOCK)
		8F	3C	00050	MOVZWL	#2328, R0	
		04	00055	2\$:	RET		

: 0722
: 0724
: 0726

; Routine Size: 86 bytes, Routine Base: \$CODE + 0283

RENAME
V04-000

VAX/VMS RENAME Utility
ERROR_HANDLER - Condition handler

H 11
16-Sep-1984 00:25:56
14-Sep-1984 12:08:48

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]RENAME.B32;1

Page 23
(8)

: 734 0727 1 END
: 735 0728 1
: 736 0729 0 ELUDOM

! End of module RENAME

.EXTRN LIB\$SIGNAL, LIB\$STOP

PSECT SUMMARY

: Name Bytes Attributes
: \$CODE 729 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

: File Total Symbols Loaded Percent Pages Mapped Processing Time
: _\$255\$DUA28:[SYSLIB]STARLET.L32;1 9776 41 0 581 00:01.0

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:RENAME/OBJ=OBJ\$:RENAME MSRC\$:RENAME/UPDATE=(ENH\$:RENAME)

: Size: 684 code + 45 data bytes
: Run Time: 00:14.9
: Elapsed Time: 00:50.0
: Lines/CPU Min: 2931
: Lexemes/CPU-Min: 19025
: Memory Used: 103 pages
: Compilation Complete

0051 AH-BT13A-SE
VAX/VMS V4.0

**DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY**